REMARKS/ARGUMENTS

In view of the foregoing amendments and the following remarks, reconsideration of this application is requested. Claims 1-21 are now pending with claims 1, 9, 15, 18, and 21 being independent. Claims 1, 9-10, 13-15, 17-18, and 20-21 have been amended.

The abstract of the disclosure has been amended in response to the Examiner's objection. The title of the invention has been amended to make it clearly indicative of the invention to which the claims are directed.

The specification has been amended to provide the serial numbers of the coassigned applications or patent numbers of issued U.S. patents and to remove attorney docket numbers. The specification has been further amended to correct grammatical errors and minor informalities. The specification and drawings have been amended in response to the Examiner's objections in paragraphs 6, 7, and 8 of the Office Action. No new matter has been introduced.

Claims 13, 14, 17, 18, and 20 have been amended in response to the Examiner's objections in the Office Action.

Claims 1, 9, 15, and 18 have been amended in response to the Examiner's rejections under 35 U.S.C. § 102(a).

As amended, claim 1 describes a digital processing system comprising a microprocessor, wherein the microprocessor performs a method for calling a subroutine that includes a number of steps. The method includes branching to the subroutine by executing a first instruction to provide an address of the subroutine. The method for calling a subroutine also includes calculating a return address by executing a second instruction that is sequentially adjacent to the first instruction to determine a relative return address.

Amended claim 9 describes a digital processing system comprising a microprocessor, wherein the microprocessor performs a method for forming a relative address that includes a number of steps. The method includes fetching a sequence of instructions in response to address locations provided by a program counter. The method for forming a relative address also includes executing a first instruction immediately after executing a branch to subroutine instruction in the sequence of instructions by using a first address value provided by the program counter as a source operand, wherein the first instruction forms the relative address.

Amended claim 15 describes a method for calling a subroutine in a digital processing system that includes a microprocessor. The method includes the step of branching to the subroutine by executing a first instruction to provide an address of the subroutine. The method also includes calculating a return address by executing a second instruction that is sequentially adjacent to the first instruction to determine a relative return address.

Amended claim 18 describes a method for forming a relative address in a digital processing system comprising a microprocessor, the method including a number of steps. The method includes fetching a sequence of instructions in response to address locations provided by a program counter. The method for forming a relative address also includes executing a first instruction immediately after executing a branch to subroutine instruction in the sequence of instructions by using a first address value provided by the program counter as a source operand, wherein the first instruction forms the relative address.

Independent claims 1, 9, 15, and 18 stand rejected under 35 U.S.C. § 102(a) as anticipated by Applicant's Admitted Prior Art (AAPA). Applicants request reconsideration and withdrawal of these rejections for at least the reason that AAPA does not describe or suggest calculating a return address by executing <u>a second instruction</u> that is <u>sequentially adjacent</u> to the first branch instruction to determine a relative return address.

AAPA, in relevant part, as shown in Figure 5 and taught on page 20, lines 17-26 describes a DSP that executes a branch instruction and then forms a return address by executing a move constant instruction (MVK) 502 that moves the least significant half of the address "LABEL" into the least significant half of general purpose register B3. The DSP next executes a move constant instruction high (MVKH) 502 that moves the most significant half of the address "LABEL" into the most significant half of general purpose register B3. AAPA does not describe or suggest calculating a return address by executing a second instruction that is sequentially adjacent to the first branch instruction. In AAPA, as shown in Figure 5, the MVKH instruction that forms the return address 503 is not sequentially adjacent to the branch instruction but follows MVK instruction 502. AAPA also does not describe or suggest <u>calculating</u> a return address by executing <u>a second instruction</u> to determine a relative return address. AAPA teaches the use of two instructions to form a return address, with the instructions each moving half of the

return address to a general purpose register. For at least these reasons, Applicants respectfully submit that claims 1, 9, 15 and 18 are patentable over AAPA.

Claims 4, 7; 10; and 19 depend from independent claims 1, 9, and 18, respectively. Accordingly, Applicants request reconsideration and withdrawal of the rejections for claims 4, 7, 10, and 19 for the reasons discussed above with respect to claims 1, 9, 15, and 18.

Independent claims 1 and 15 stand rejected under 35 U.S.C. § 102(b) as anticipated by Iwao (4,799,151). Applicants request reconsideration and withdrawal of these rejections for at least the reason that Iwao does not describe or suggest calculating a return address by executing a second instruction that is sequentially adjacent to the first branch instruction to determine a relative return address.

Iwao, in relevant part, as shown in Figure 5 and taught in the Abstract describes a branch and link (BAL) instruction 520(600) that calls a subroutine SUB-S 700. The BAL instruction causes a return address 521(601) from which to restart instruction execution in the main program after completion of the subroutine to be stored on a stack. A return instruction RTN 730 that is executed at the end of subroutine 700 causes the return address 521(601) stored on the stack to be loaded into the program counter and the main program restarts execution at instruction 521(601) specified by the return address. As shown in 750, an offset value α=5 from the return address in the main program may be designated. Thus, the return address in the main program and the offset value can be added to restart instruction execution in a different portion 526(606) of the main program. Iwao does not describe or suggest calculating a return address by executing a second instruction that is sequentially adjacent to the first branch instruction to determine a relative return address. In Iwao, the return instruction 730 or 750 occurs at the end of the subroutine 700 after a sequence of other instructions have been executed by the processor. Thus, in Iwao, a number of instructions separate the branch instruction BAL from the return instruction and the return instruction that determines the return address is not sequentially adjacent to the first branch instruction BAL. For at least these reasons, Applicants respectfully submit that claims 1 and 15 are patentable over Iwao.

Claims 2-3, 6-7; and 16-17 depend from independent claims 1 and 15, respectively. Accordingly, Applicants request reconsideration and withdrawal of the rejections for claims 2-3, 6-7, and 16-17 for the reasons discussed above with respect to claims 1 and 15.

Dependent claims 5 and 12 and independent claim 21 stand rejected under 35 U.S.C. § 103(a) as obvious over AAPA in view of Hennessy and Patterson, Computer Architecture – A Quantitative Approach, 2nd Edition, 1996 (Hennessy). However, Hennessy does not remedy the failure of AAPA to describe or suggest calculating a return address by executing a second instruction that is sequentially adjacent to the first branch instruction to determine a relative return address. Hennessy shows in Figure 3.1 of page 130 the circuitry for a pipelined datapath with a plurality of stages. On page 134 in Figure 3.4, Hennessy describes pipelining the datapath by adding a set of registers between each pair of pipe stages. The caption to Figure 3.4 describes carrying the PC information from IF/ID register to MEM/WB register through the stages of the pipeline. Figure 3.22 on page 163 shows an alternative pipeline for the datapath of Figure 3.4 in which pipeline stalls are reduced by moving the zero test and branch target calculation into the ID/EX phase of the pipeline from the EX/MEM phase of the pipeline. Hennessy on pages 168-169 describes different schemes to reduce performance penalties caused by branch instructions. In delayed branching, sequential successors are executed whether or not the branch is taken. Hennessy's DLX architecture has one delay slot after each branch instruction shown in Figure 3.27 in which a Branch-delay instruction ($I + 1$) is executed. In Hennessy's architecture, as shown in Figure 3.28 an optimizing compiler can schedule an instruction for the delay slot after examining the dependencies of instructions. Thus, Hennessy does not describe or suggest calculating a return address by executing a second instruction that is sequentially adjacent to the first branch instruction to determine a relative return address.

Calculating a relative return address by executing a second instruction that is sequentially adjacent to the first branch instruction has several advantages. Use of a single second instruction reduces code size and power dissipation since fewer instructions are fetched. Another advantage as shown in Figure 6A is that the second instruction ADDKPC provides a relative return address so that the program sequence 600 and subroutine 610 may be relocated to another program address range without changing program sequence 600.

For at least the reasons given above, Applicants respectfully submit that claims 5, 12, and 21 are patentable over AAPA in view of Hennessy.

Amended claim 21 describes a method for operating a compiler for a microprocessor, wherein the microprocessor executes a first type of instruction that performs a specified

operation and also directs that a selectable number of virtual no-operation (NOP) instructions be executed after executing the first type instruction. The method includes the steps of determining that a first instruction of the first type of instruction is to be executed in a delay slot of a first branch type instruction, wherein the first instruction calculates a first return address. The method also includes determining that a second instruction of a second type of instruction is to be executed in a delay slot of the first branch type instruction, wherein the second instruction calculates a second return address.

Independent claim 21 stands rejected under 35 U.S.C. § 103(a) as obvious over AAPA in view of Hennessy. The Examiner in paragraph 37 of the Office Action mailed October 20, 2003 referring to AAPA states that:

> a) it has been determined that a first instruction of the first type of instruction is to be executed in a delay slot of a first branch type instruction. See Fig.5, instruction 503.
> b) it has been determined that a second instruction of a second type of instruction is to be executed in a delay slot of the first branch type instruction. See Fig.5, instruction 502.

Applicants request reconsideration and withdrawal of these rejections for at least the reason that AAPA does not describe or suggest a first instruction calculating a first return address and a second instruction calculating a second return address.

As described above, AAPA, as shown in Figure 5 describes a DSP that executes a branch instruction and then fotms a return address by executing a move constant instruction (MVK) 502 that moves the least significant half of the address "LABEL" into the least significant half of general purpose register B3. The DSP next executes a move constant instruction high (MVKH) 502 that moves the most significant half of the address "LABEL" into the most significant half of general purpose register B3. AAPA does not describe or suggest a first instruction calculating <u>a first return address</u> and a second instruction calculating <u>a second return address</u>. AAPA describes moving a single address "LABEL" to general purpose register B3 using two instructions. Nor does Hennessy, as mentioned above, remedy the failure of AAPA to describe or suggest a first instruction calculating a first return address and a second instruction calculating a second return address. Accordingly, Applicants request reconsideration and withdrawal of the rejection of claim 21 for at least this additional reason and the reasons discussed above with respect to claims 1, 9, 15, and 18 and claims 5, 12.

Dependent claims 8, 11, and 20 stands rejected under 35 U.S.C. § 103(a) as obvious over AAPA in view of Texas Instruments, TMS32010 User's Guide, 1983 (TI). However, TI does not remedy the failure of AAPA to describe or suggest calculating a return address by executing a second instruction that is sequentially adjacent to the first branch instruction to determine a relative return address. TI shows an Instruction Set Summary in Table 3-2 describing the function of various instructions. For example, the LTA instruction is described as "LTA combines LT and APAC [instructions] into one instruction". TI does not describe or suggest any instructions in Table 3-2 that calculate a relative return address and are sequentially adjacent to the first branch instruction. Accordingly, Applicants request reconsideration and withdrawal of the rejection for at least the reason discussed above and with respect to claims 1, 9, 15, and 18.

Applicants respectfully traverse the rejections of claims 8, 11, and 20 under 35 U.S.C. § 103(a) for the following additional reasons. The claim contains patentable features not taught or suggested in either of the cited art. To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). See MPEP 2143.03. Applicants asserts that the cited art does not describe or suggest all of the claim limitations.

For example, claims 8, 11, and 20 recite executing a second instruction that specifies a number of virtual no-operation (NOP) instructions. As recited in claim 1, 9, 15, and 18 the second instruction is used to calculate a return address. Applicants can find no teaching of the claimed limitation from claims 8, 11, and 20 in AAPA or TI. The Examiner in paragraphs 39, 40, and 41 of the Office Action mailed October 20, 2003 states that "TI has taught the concept of combining multiple instructions into a single instruction" and gives the example of the LTA instruction. However, Table 3-2 from the TI reference does not describe or suggest a NOP instruction or any instruction that as part of its functionality specifies a number of virtual no-operation instructions. Thus, Applicants respectfully suggest that it would not have been obvious to one of ordinary skill in the art to have the second instruction specify the number of virtual no-operation instructions. For at least this reason, Applicants respectfully submit that the rejection of claims 8, 11, and 20 under 35 U.S.C. § 103(a) is improper and that the claim is patentable over AAPA in view of TI.

Dependent claim 13 stands rejected under 35 U.S.C. § 103(a) as obvious over AAPA in view of Hennessy and further in view of Sharangpani et al. (6,237,077). However, Sharangpani does not remedy the failure of AAPA and Hennessy to describe or suggest calculating a return address by executing a second instruction that is sequentially adjacent to the first branch instruction to determine a relative return address. Sharangpani, as described in the Abstract, teaches use of an instruction bundle for processing one or more branch instructions. The instructions are ordered in an execution sequence within the bundle, with the branch instructions ordered last in the sequence. The first branch instruction in the bundle that will be taken is determined and subsequent instructions in the execution sequence are suppressed. Sharangpani does not describe or suggest calculating a return address by executing a second instruction that is sequentially adjacent to the first branch instruction since in Sharangpani subsequent instructions after the first branch instruction are suppressed. Accordingly, Applicants request reconsideration and withdrawal of the rejection for at least the reasons discussed above and with respect to claim 9.

Dependent claim 14 stands rejected under 35 U.S.C. § 103(a) as obvious over AAPA in view of Haataja (6,137,836). However, Haataja does not remedy the failure of AAPA to describe or suggest calculating a return address by executing a second instruction that is sequentially adjacent to the first branch instruction to determine a relative return address. Haataja describes in his Abstract a method of transmitting pictorial data resulting in reduction of required transmission bandwidth by constructing the pictorial data in the form of a composite image of primitive pictures. Figure 8 shows construction of the composite image at a source of pictorial data, and communication of the pictorial data to a portable communicator. The portable communicator may be a paging device, PDA, cellular telephone modified for regenerating the composite image and presenting the composite image on a display. Haataja does not describe or suggest calculating a return address by executing a second instruction that is sequentially adjacent to the first branch instruction to determine a relative return address. Accordingly, Applicants request reconsideration and withdrawal of the rejection for at least the reasons discussed above and with respect to claim 9.

In view of these remarks and amendments, Applicants submit that this application is now in condition for allowance and the Examiner's prompt action in accordance therewith is

respectfully requested. The Commissioner is authorized to charge any additional fees and/or credit any overpayment to Deposit Account 20-0668 of Texas Instruments Incorporated.

Respectfully submitted,

Indranil Chowdhury

Indranil Chowdhury
Attorney for Applicants
Reg. No. 47,490

Robert D. Marshall, Jr.
Texas Instruments Incorporated
P.O. Box 655474, MS 3999
Dallas, TX 75265
(972) 917-5290

Attachments